LLNL-CONF-413253

LAWRENCE LIVERMORE NATIONAL LABORATORY

A Case for Database Filesystems

P. A. Adams, J. C. Hax

May 22, 2009

Space Mission Challenges for IT - 2009
Pasadena, CA, United States
July 19, 2009 through July 23, 2009

A Case for Database Filesystems

Philip A. Adams, National Ignition Facility
John C. Hax, Oracle Corporation, Member IEEE

Abstract— **Data intensive science is offering new challenges and opportunities for Information Technology and traditional relational databases in particular. Database filesystems offer the potential to store Level Zero data and analyze Level 1 and Level 3 data within the same database system [2]. Scientific data is typically composed of both unstructured files and scalar data. Oracle SecureFiles is a new database filesystem feature in Oracle Database 11g that is specifically engineered to deliver high performance and scalability for storing unstructured or file data inside the Oracle database. SecureFiles presents the best of both the filesystem and the database worlds for unstructured content. Data stored inside SecureFiles can be queried or written at performance levels comparable to that of traditional filesystems while retaining the advantages of the Oracle database.**

***Index Terms*—High Availability, Database Systems, Data Management, Information Systems, Data Intensive Science, eScience**

I. Introduction

"Show me a scientist who did not want more data," said Alex Szalay of John Hopkins University [1]. Certainly the statement is true for planetary science, but it is also almost certainly true for all scientists and all scientific disciplines.

Two unique trends are currently occurring within scientific computing. The first trend is the explosion in data volumes which is being driven by a number of factors: better and more diverse instrumentation, flexible optics, coordinated multi-instrument observatories, and in a self perpetuating cycle, improvements in Information Technology (IT) itself. In addition, international funding efforts have enabled larger, more complex instruments. The second trend in scientific computing is the increasingly collaborative nature of data analysis which is driven by the funding model, the global distribution of the scientific knowledge base, and the realization that the collaborative model is often more productive than research at the individual level. The philosophy of the NASA's Science Mission Directorate (SMD) is that science does not result from the launch of a mission or the collection of data. Rather, science only occurs through the analysis and understanding of that data. Broadly defined, Research & Analysis is the process used by NASA to produce the concept studies that provide the scientific basis for a mission, the necessary technology and techniques for implementing the mission, the calibration, validation, and analysis of data as a mission is underway, and the analysis of archived data after a mission ends. Healthy mission science teams and data analysis programs must accompany every successful mission. Scientists have noted the marked disparity between tremendous growth in the performance of computers, sensors, data storage, networks, and other system elements, and the decidedly slower growth in scientific insight. Furthermore, they assert that this disparity is due, in part, to the increasing complexity of managing ever larger and more distributed computations and data. [3]

With diagnostic equipment, satellite, video, and camera imagery streaming data at ever increasing rates, it is now commonplace for large scientific projects to generate petabytes of data. SQL/Scalar, XML, Image, Monte Carlo Simulations, Audio/Video, telemetry, and spectrometers are examples of the diverse types of data generated or utilized by space missions. Vector and spatial data is then available for scientific analysis. The Lunar Reconnaissance Obiter (LRO) is an example of a platform that generates multiple data types. LRO will generate standard gif images, streaming video, spectrometry data, photometer data, and telemetry data. However, many traditional data analysis and management approaches have split structured and unstructured data across databases, file and email servers, workstations, and laptops. This distribution not only compromises the security and integrity of the data, but more importantly makes determining its pedigree harder. In addition, searching and data integration are time-consuming if not difficult tasks in the distributed environment. There is huge potential for scientific discovery

Manuscript received April 28th, 2009. This work was in part performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344 and in part performed under the auspices of the Oracle Corporation.

Philip A. Adams is a Sr. Systems Architect with the National Ignition Facility at Lawrence Livermore National Lab, Livermore, CA. (925-422-0493; adams64@llnl.gov.).

John C. Hax is a Sr. Systems Architect with Oracle Corporation, 7604 Technology Way, Denver, CO 80237 (john.hax@oracle.com; 720-979-3220.

by combining information from these multiple, diverse and distributed data resources.
Traditionally, filesystems have been the primary method of storing scientific data and images. Relational database systems (RDBMS) were primarily used for the storage of metadata. This has the negative impact of relegating scientific data from structured to semi-structured file characteristics. There are numerous drawbacks to having content stored in an unstructured manner. These drawbacks include but are not limited to data curation, pedigree, security, availability, recoverability, and manageability. It should also be mentioned that a reliance on filesystems often precludes the utilization of institutional IT resources and leveraging institutional investments in commercial software. Many of the data challenges faced by the high energy density (HED) physics community are similar to those faced by the astronomy and planetary science community. As a leader in innovation and HED physics, the National Ignition Facility (NIF) has embraced database filesystems as a way to bridge the gap between relational databases and filesystems.

II. Filesystems

Historically, filesystems have been the preferred method for storage of unstructured content and have provided better throughput directly to storage devices for Tier 0 (raw experimental data) content. Examples of distributed architectures that use filesystems for both structured and unstructured content are HDF5, Lustre [5], and Google Filesystem (GFS). These solutions aim to provide maximum scalability to meet data volume and ingestion requirements with provisions for fail over and high availability through data replication.

Another heralded advantage of filesystems is the ubiquity of accessing a filesystem. A wide variety of operating systems support protocols such as NFS, SMB, CIFS and FTP. By extension, many application programming interfaces are available to do standard file manipulation such as file open (f_open), file close (f_close), importing the java io package, or using ifstream/ofstream C++ file I/O classes.

Conversely, data stored in legacy relational databases could only be accessed via SQL using application specific database drivers. Although there has been wide support for database drivers by operating systems and programming libraries to make database data equally accessible, these access mechanisms have not been able to completely replace filesystem access mechanisms. Because no common database and filesystem access protocol was available, the burden shifted to application developers and scientific researchers to make sense of the two silos of information.

A. Map Reduce

The MapReduce programming model popularized at Google, Yahoo!, and Facebook has triggered a lot of debate about the best way to analyze data. By utilizing this framework, certain kinds of problems can be distributed across a large server farm. It is not uncommon for a server farm running Hadoop or Google Filesystem (GFS) to process many 100's of GBs of data in hours. This capability further propagates the notion of using a filesystem to store and analyze data.

Despite the ability to rapidly analyze an impressive amount of data, query response times for more targeted analysis or interactive querying can be dismal compared to an RDBMS [11]. Furthermore, as a data management, data curation, and collaborative engine, MapReduce falls short. Hence its overall usefulness has been scrutinized by many database luminaries such as Michael Stonebraker and David DeWitt [12]. In an effort to bridge the gap between the two data analysis technologies, the Hive project, a data warehouse infrastructure built on top of Hadoop, was created to enable a SQL based query language to be run. Many advocates of the MapReduce paradigm suggest that it is not a database system, so don't judge it as one. Rather, accept it for the power tool that it is.

The MapReduce framework is implemented in C++ with interfaces in Python and Java. A scientist has to be a programmer – not just a data consumer – to use it. This runs counter to the stated goals of scientific institutions, which is to increase the amount of time that researchers spend on science. [4]

Because the technical advantages of MapReduce are difficult to ignore, one could guess that future databases will incorporate a powerful SQL engine with MapReduce functionality. This could potentially solve the issue often experienced in petabyte-scale data analysis where significant time is wasted transferring data to the cluster for processing. The need is intensifying to unify analytical processing and curation while providing a common platform for queries, machine learning, text mining, and statistical computing.

III. Modern Databases: Relational Data Meets Files

Prior to the latest generation of database engines, the sheer number of data types, complexity and diversity made combining data in any single repository a daunting task with many technical challenges. In addition, solutions were not available that were well matched to the science requirements or the scalability challenges that large data volumes present. This forced a number of scientific projects to split the data between a database and a filesystem or use the filesystem entirely for all data types.

However, the split data model causes a number of data management issues. Without a single source of truth for highly dynamic data, there is burden keeping disparate data repositories current. Disjointed security and auditing rules and a fragmented backup and recovery policy are also common challenges for a split data model. Merging data between disparate data silos can be a time consuming task. For large volumes, transporting data through the network introduces undue latency on dissemination of critical analysis results.

Modern-day RDBMS databases, such as Oracle's 11g Database Server, can accommodate the disparate data types and provide a framework enabling more ambitious data integration architectures that are well-adapted for semantic grids, simulation, analysis, data mining, and visualization. SQL, PL/SQL, Java, C, and PHP are available to load, search, join, compute and display results. In addition, the database eases the burden of data management by providing a security infrastructure, methods for holistic backup and recovery, and long term archival and retrieval of historical data.

A. Relieving the Burdens of Long Term Data Curation

Data curation is seldom a primary focus of many scientific endeavors. However, in order to enable collaborative science, other research communities and projects need to be able to access those repositories and be able to validate the pedigree, and therefore the accuracy, of the data. Without sound data stewardship practices, many future systems will be unable to use the data. Storage systems holding the data may be unreliable. Many scientists in the HED realm have found it cumbersome to access research data on legacy cartridges and tapes. Searching for specific data may be next to impossible without undertaking a huge data migration project – and even then, the data integrity may be in question due to inadequate data models and management techniques. By keeping the data in a robust repository such as an Oracle database, as storage subsystems change, the database can transparently move the data from one media type to another using Automatic Storage Management (ASM).

Level 0 (Raw) data is typically transformed and enriched with data from disparate systems. What happens when a diagnostic is found to have incorrect calibration data? Without strict relationships, this could be a nightmare. It may be easier to rerun analysis to reproduce the Level 1, 2 and 3 data. However, an unknown quantity of Level 4 content has been generated from this data and is stored on many researchers' workstations and file shares. By utilizing a content management framework suited for eResearch, the observation data that validates the research activity is preserved and cataloged with a Uniform Resource Name (URN) along with links to reference material. When an update of data occurs that underpins the logic of a higher level document, automatic invalidation of the document can occur as part of the workflow until the both the data and the document have been reviewed. By properly curating data and using systems that enable this activity, future research communities and projects will be able to leverage the information stored in the virtual data museum for years to come.

B. Metadata Management

Metadata is "data about data." Filesystems store provide very simple metadata on a per file or directory basis such as author, file creation, modification time, last accessed, file size, and permissions. A form of rudimentary user-defined metadata might be the naming and folder conventions.

The existence of these user-defined conventions indicates there is a requirement for far richer metadata to describe content. In addition to the common metadata, content management systems support the definition of custom metadata attributes by content type. For example, in the NIF environment, every optics inspection image has attributes for pixel height and width, orientation, camera location, and resolution. These data attributes are needed by the software to interpret and process these images.

C. Data Access Methods and Standards

One of the greatest challenges for many scientific collaboration projects, including the National Virtual Observatory (NVO) has been the establishment of data access methods and standards [13]. The basic goal for any data repository is to get the data off the disks and hand it to the requester in a usable format. Therefore, the need to support a variety of access protocols is very important as researchers challenge the data repository with a variety of tools. For some time, Oracle databases have supported WebDAV, SQL, PL/SQL (Oracle's Procedural Language), OCI, OCCI, and Java. With Oracle 11g SecureFiles, the access methods have been extended - allowing both database clients and filesystem clients to access the same content. Apart from being SQL standards compliant, SecureFiles provides POSIX compliant filesystem interfaces and content can be accessed through open data protocols such as HTTP, NFS, and FTP. By embracing common standards in the traditional database and filesystem worlds, modern databases are well suited to be at the center of all scientific data analysis efforts.

D. Web Ontology Language: A new way to analyze data

Modern databases, in addition to providing new access protocols, support additional mechanisms for data analysis. As an example, the OWL Web Ontology Language is designed for use by applications that need to process the content of information instead of just presenting information to humans. The OWL Web Ontology Language is intended to provide a language that can be used to describe the classes and relations between content that are inherent in Web documents and applications. OWL facilitates greater machine interpretability of Web content than that supported by XML, RDF, and RDF Schema (RDF-S) by providing additional vocabulary along with formal semantics. OWL has three increasingly-expressive sublanguages: OWL Lite, OWL DL, and OWL Full [7]. RDF data types are queried by the SPARQL query language. Both OWL and SPARQL are W3C standards. Oracle offers full support for both SPARQL and OWL. These technologies offer great promise for finding new ways to analyze data in both consolidated and distributed data repositories. The United Kingdom's National Mapping Agency, Ordnance Survey, uses the Semantic Web internally to more accurately and inexpensively generate geographic maps. OWL and the Semantic Web are predicated upon tagging and marking up data. It is much easier to mark up data at the meta-data level than when that data is contained in flat files and stored in filesystems. Currently, scientific researchers are developing some of the most advanced applications, including a system that pinpoints genetic causes of heart disease and another system that reveals the early stages of influenza outbreaks [8].

IV. National Ignition Facility's Experience using SecureFiles

A. *Overview of NIF*

The National Ignition Facility, the world's largest laser, is located at Lawrence Livermore National Laboratory in Livermore, California. When NIF's 192 laser beams focus their energy on a BB-sized target inside the target chamber, temperatures of more than 100 million degrees and pressures more than 100 billion times the Earth's atmosphere can be achieved. The mission of NIF is to become a premier international center for experimental science. In addition to enabling stockpile stewardship and high energy density research, NIF will allow scientists from around the world to gain new insights into astrophysical phenomena such as supernovae, giant gas planets and black holes.

The National Ignition Facility uses Oracle's 11g Database and its unique features for managing, storing and analyzing massive amounts of data and images resulting from the facility's large-scale experiments on inertial confinement thermonuclear fusion.

B. *NIF's use of the Database for Capture, Curation, Archiving and Preservation of data*

An Oracle 11g Database supports NIF's laser control system, which allows NIF scientists to adjust the settings of the laser and target to their exact specifications prior to their experiments, and then captures the rush of data following each experiment. Thirty minutes following the target shot, the resulting large high-resolution images and data are stored in the database in their native formats using Oracle SecureFiles, and an initial analysis is conducted to provide scientists with an overview of the experiment results.

Two types of analysis are done on a regular basis:
Target Diagnostic Analysis
Optics Inspection

During Target Diagnostic Analysis, data from a number of detectors, oscilloscopes, interferometers, streak cameras and other instruments are analyzed to measure the performance of the target. HDF files are ingested into the Oracle Database, a URN is given, and metadata is used to enrich the content object. A variety of IDL[14] scripts analyze the data contained in the files and store the results and associated metadata in the database. Pedigree metadata accompanies the results to substantiate their lineage.

During Optics Inspection, a number of raw images from a variety of cameras provide detailed information about the largest optical instrument ever built. As the inspection process analyzes images for micron-sized defects, detections are stored in the database for correlation with past defects. The results of this analysis determine whether to perform the next laser shot or whether maintenance is needed. By correlating past and present inspections in the database, damage site evolution, and other interesting optical phenomena is discovered and studied.

The intention is to keep all experimental data and analysis results available for instant retrieval throughout the approximately 30 year lifetime of the facility. Data sizes for the metadata, experimental and modeling data over a 3-4 year timeframe could easily reach the multi-petabyte range. By using the database to archive least recently used data to tape, NIF is able to keep its spinning disk footprint down while still preserving the ability to retrieve data transparently to the user whether the data is seconds or decades old.

- Architecture involved:
  - Oracle 11gRAC with SecureFiles
  - Oracle CMSDK
  - Oracle BPEL
  - Java, IDL, ImageMagic

V. Oracle SecureFiles Database Filesystem

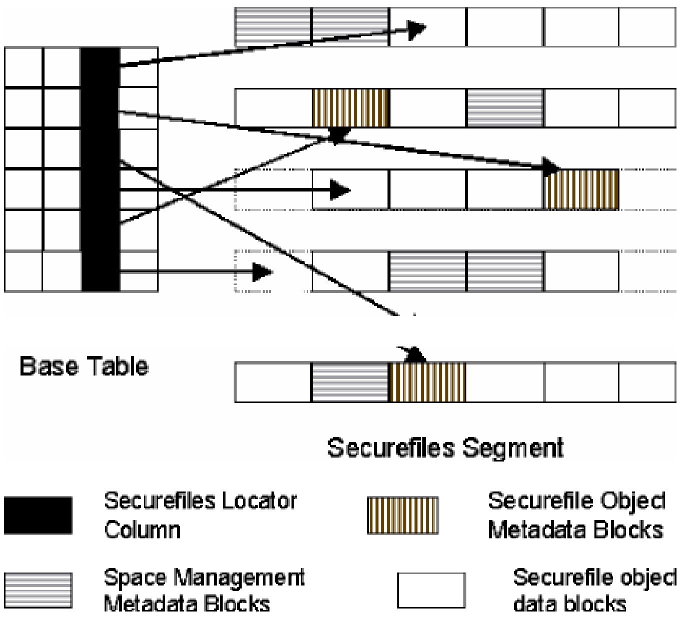
A. *Overview*

Oracle SecureFiles is a novel architecture that provides the scalability of filesystems with the rich features and benefits associated with the Oracle database. Prior to SecureFiles, database systems were deemed too slow to accommodate the I/O rates required for ingestion of large numbers of images and experimental results. SecureFiles stores data as a first class object within the database and supports all types of content without compromising throughput or scalability. Oracle SecureFiles extends data atomicity, consistency, and durability from metadata to the previously unstructured content stored as B-Files in a filesystem. Metadata can be extended to include file type, pixel count, size, and many other characteristics of the data that simplify search and analysis and streamline data loading and parsing. In short, Oracle SecureFiles treats all content equally. SecureFiles delivers or exceeds filesystem performance for basic read/writes and provides better scalability than traditional filesystems. In addition to content equality, SecureFiles extends filesystem features such as de-duplication and advanced filesystem compression to optimize utilization of cache and storage. If leveraged, these advantages stand to enable and reduce the cost of Research & Analysis in support of space missions.

B. Architecture

The structural design of SecureFiles is similar to that of filesystems. Unstructured data associated with semi-structured content is associated and stored as a SecureFile object. A SecureFile object is a collection of dynamic chunks allocated from and stored in the Oracle database. Each chunk is a set of contiguous data blocks as shown in Figure 1.

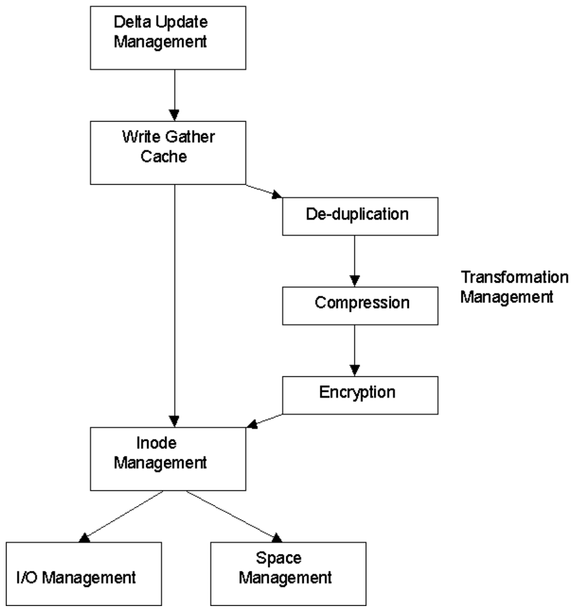
Figure 1. Base Table – Oracle table holding metadata plus locator columns similar to a b-file pointer. [10]


Base Table
Securefiles Segment
Securefiles Locator Column
Securefile Object Metadata Blocks
Space Management Metadata Blocks
Securefile object data blocks

C. Components

There are 6 major components comprising the SecureFiles architecture. These are: Delta Update, Write Gather Cache, Transformation management, Inode Management, Space management, and the IO Management. Each SecureFile segment is a collection of database extents that are contiguous data blocks. This segment is a free space pool. Logically, a SecureFiles segment consists of blocks that contain metadata for space management and blocks that are SecureFile objects. These components are shown in figure 2.

Figure 2. Components of SecureFiles



(1) Delta Update – Oracle SecureFiles introduces the concept of ‘delta updates’, which enables non length-preserving updates of file-like objects without undergoing file updates or re-writes of existing data blocks. This is a major differentiator with respect to filesystems or LOB’s. Updates to objects in filesystems as well as databases require re-writes of portions of objects that preserve length of the updates. This causes inefficiencies in IO even for small updates. The Delta Update component provides API’s that specify the object to update, mapping of source and destination offsets, length of the delta, and ensures IO cost of update operations is linear to the size of the delta. The delta update operation also provides substantial benefits in the performance of XML storage frameworks.

(2) Write Gather Cache (WGC) – the write gather cache is a subset of the database buffer cache. The WGC gather buffers a user specified amount of data before flushing to the storage layer. This buffering of in-flight data greatly optimizes IO by allocating larger disk tracts.

(3) Transformation Management (TM) - TM is comprised of three major components; deduplication, compression, and encryption. Cumulatively, these three features can greatly reduce the cost and complexity of long-term information curation and storage.

- Deduplication -The Oracle database server automatically detects duplicate copies by generating both a pre-hash and a full hash. eScience tends to generate large amounts of images. Deduplication can greatly simplify image management by simply maintaining a pointer to the base image. Often it is the delta between image metadata that is more important than the actual image itself.

- Compression – Oracle automatically detects if SecureFile object data is compressible and compresses using multiple file compression algorithms. If compression does not yield any space savings or the data is already compressed, SecureFiles will automatically turn off compression.

- Encryption – Encryption is a component of many NASA and JPL funded missions. Oracle SecureFiles uses Transparent Data Encryption (TDE) syntax for encryption. Oracle SecureFiles supports automatic key management.

(4) Inode Management – The inode management layer is responsible for initiating on-disk storage and access operations on SecureFiles data in the buffer layer. Based on the array of chunks returned by the space management layer, the inode manager stores the either in the row-column intersection of the base table associated with the object, or in the most current header block of the SecureFile object.

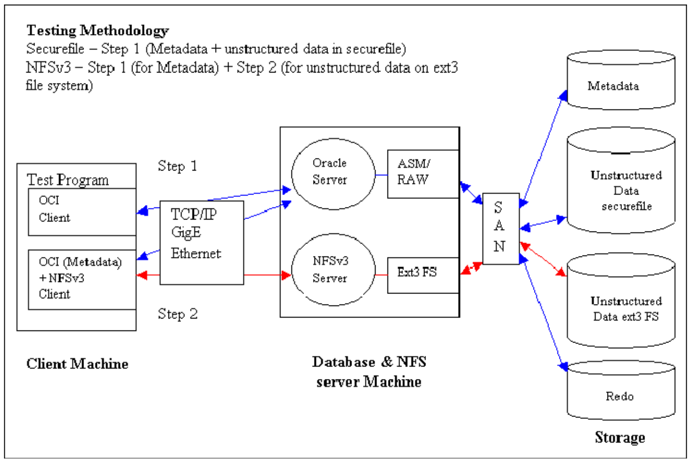
(5) Space Management – The space management layer is responsible for allocating free disk space to SecureFile objects and de-allocating used space from SecureFile objects to the SecureFile segment on disk. The space management layer also support allocation of variable sized chunks.

(6) I/O Management – The I/O Management Layer is responsible for satisfying I/O requests during reads and writes of SecureFile objects. During writes, the Inode Manager communicates the set of chunks obtained from the space management layer and the write gather cache buffers to the I/O Manager. The I/O Manager either writes to the write buffer cache or asynchronously writes to the disk. Prior to the disk write, the I/O Manager coalesces the chunks to optimize disk throughput.

D. Performance

Oracle has performed benchmarks of SecureFiles against NFS v3 filesystems in the environment described in Figure 3.

Figure 3. Architecture of the insert-only experiment for image and video data


Testing Methodology
Securefile – Step 1 (Metadata + unstructured data in securefile)
NFSv3 – Step 1 (for Metadata) + Step 2 (for unstructured data on ext3 file system)
Test Program
OCI Client
OCI (Metadata) + NFSv3 Client
Step 1
TCP/IP GigE Ethernet
Step 2
Client Machine
Oracle Server
ASM/ RAW
NFSv3 Server
Ext3 FS
Database & NFS server Machine
S A N
Metadata
Unstructured Data securefile
Unstructured Data ext3 FS
Redo
Storage

Insert-only experiments were performed for both image and video data on hardware similar to Figure 3. One difference is that to minimize network bottlenecks the server is both the client and server. Concurrency varied from 1-16 processes. Throughput numbers are shown in Figure 5 and 6.

WORKLOAD DETAILS

The test application simulates a real world DICOM application consisting of patient metadata and DICOM images [15]. We compare the performance of SecureFiles to that of the NFSv3 filesystem. In both cases, metadata is stored in an Oracle database. In the case of filesystem, the DICOM images are stored on file servers that are accessed from a client machine over NFSv3. In the case of SecureFiles, the metadata as well as the DICOM images are stored inside an Oracle database. The schema used for the test is described in Figure 4.

Figure 4. Schema Description

Filesystem	SecureFile
number(10) primary key	number(10) primary key
varchar2 (300) Owner	varchar2 (300) Name
varchar2 (100) PathToFile	securefile Document

SECUREFILES VS FILESYSTEM (NFSV3 OVER EXT3 FS)

This section compares the performance of SecureFiles with the Filesystem (for the workload discussed above). ***Note that these throughput numbers are dependent on the network and storage setup***. Details of the setup can be found in the configuration section of this document.

In Oracle Database 11g, SecureFiles supports a new logging level, FILESYSTEM_LIKE_LOGGING, which is similar to logging available with popular filesystems. When SecureFiles logging is set to this level, Oracle writes only the metadata to the redo log. This setting is similar to the metadata journaling of filesystems, which reduces mean time to recovery from failures and is sufficient for crash recovery or instance recovery. SecureFiles also supports database logging in which both the metadata and LOB data are written to the redo log. This is especially useful when media recovery or standby databases are required. In such cases, archive log should be turned on. In our tests, we set SecureFiles to FILESYSTEM_LIKE_LOGGING mode to keep the logging level and functionality comparable to that of the filesystem (ext3). Note that commit guarantees data on disk.

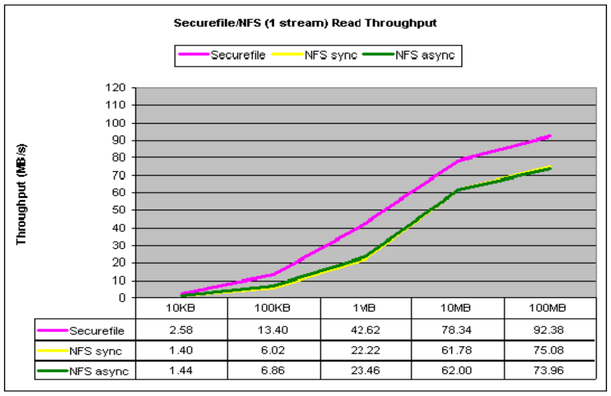
Single and multi-stream tests were done to see the performance and scalability. We have tried to use the best performing options for NFSv3 (async and rwsize of 32KB). The noatime and writeback options for ext3 were also tried but the results were not different from the NFSv3 async numbers.

Read Performance

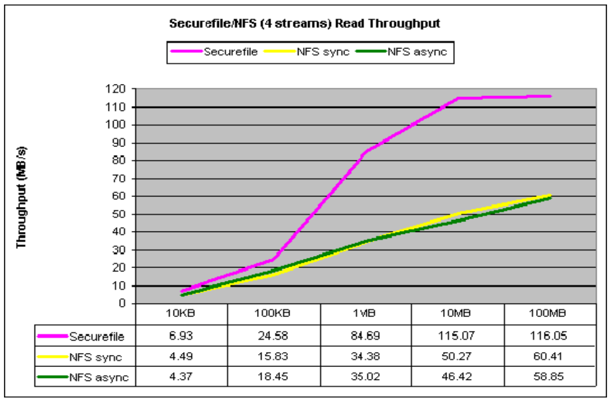
SecureFile outperforms the NFSv3 access for all sizes. Gains for the smaller file sizes are also due to reduced roundtrips where metadata and data is accessed in one roundtrip unlike the NFSv3 case where metadata and file is accessed in separate roundtrips.

This demonstrates the improvements due to intelligent pre-fetching, larger I/O sizes due to better contiguous space allocations and network optimizations.

Figure 5. Read Throughput



	10KB	100KB	1MB	10MB	100MB
Securefile	2.58	13.40	42.62	78.34	92.38
NFS sync	1.40	6.02	22.22	61.78	75.08
NFS async	1.44	6.86	23.46	62.00	73.96



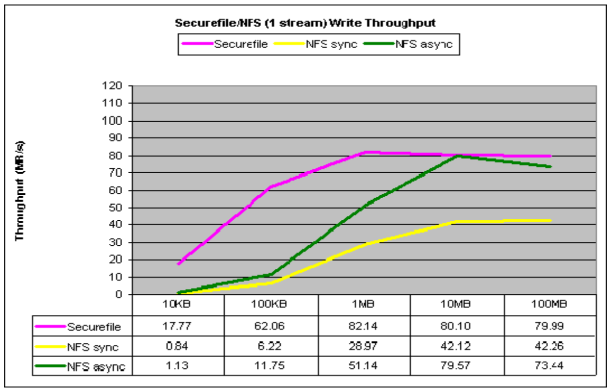
	10KB	100KB	1MB	10MB	100MB
Securefile	6.93	24.58	84.69	115.07	116.05
NFS sync	4.49	15.83	34.38	50.27	60.41
NFS async	4.37	18.45	35.02	46.42	58.85

Write Performance

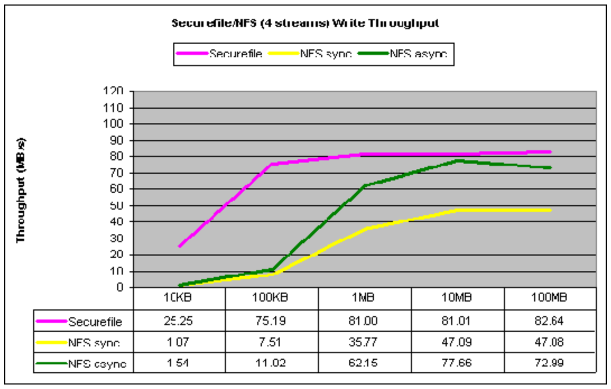
SecureFile outperforms the NFSv3 access for all sizes. Again as in the read case, for small file sizes the NFSv3 case has the overhead of writing metadata and file separately in two roundtrips.

This demonstrates the improvements due to the write gather cache, larger contiguous I/O and space pre-allocation optimizations.

Figure 6. Write Throughput



	10KB	100KB	1MB	10MB	100MB
Securefile	17.77	62.06	82.14	80.10	79.99
NFS sync	0.84	6.22	28.97	42.12	42.26
NFS async	1.13	11.75	51.14	79.57	73.44



	10KB	100KB	1MB	10MB	100MB
Securefile	25.25	75.19	81.00	81.01	82.64
NFS sync	1.07	7.51	35.77	47.09	47.08
NFS async	1.54	11.02	62.15	77.66	72.99

VI. Conclusion

The ultimate goal of science is to create new knowledge and new discoveries. We believe that database filesystems offer many advantages over traditional filesystems that will enable the scientific community to utilize tools to enable scientists to analyze data in new ways and overcome many of the challenges of data intensive science.

We have explored how programs as diverse as HED physics and space exploration can benefit from utilizing database filesystems. The National Ignition Facility has demonstrated the viability and efficiencies gained by using database filesystems and associated RDBMS features in the construction of its scientific data archive.

As new discoveries are made and data volumes increase, it is imperative to have a robust database system that is not only capable of managing the pedigree of that data, but also serve as a knowledge repository for the future. Analyzing data in new ways will lead to new discoveries.

REFERENCES

[1] Alex Szalay, Science Magazine, Vol 323, 6 March, 2009 Podcast

[2] http://podaac.jpl.nasa.gov/WEB_INFO/glossary.html#DDD

[3] Gray et al. Scientific Data Management in the Coming Decade

[4] NSF - Workshop on the Challenges of Scientific Workflows May 12, 2006 Arlington, VA

[5] http://nasascience.nasa.gov/researchers

[6] LUSTRE Filesystem: High Performance Storage and Scalable Filesystem. Sun, Dec 2007

[7] Olsen, Zimmerman, Bos –Scientific Collaboration on the Internet

[8] http://www.w3.org/TR/owl-guide/#OWLGlossary

[9] The Semantic Web in Action, Scientific American Dec. 2007

[10]Ganesh, et al. http://www.vldb.org/pvldb/1/1454170.pdf

[11]Apache HIVE http://wiki.apache.org/hadoop/Hive

[12]Stonebraker/DeWitt – Map Reduce: A major step backwards http://www.databasecolumn.com/2008/01/mapreduce-a-major-step-back.html

[13] Ackerman, Mark S., Erik Hofer, and Robert Hanisch. “Collaboratory: The National Virtual Observatory.” Gary Olson, Ann Zimmerman, and Nathan Bos (eds.), Science on the Internet, MIT Press, 2007

[14] http://www.ittvis.com/ProductServices/IDL.aspx

[15] http://medical.nema.org/

IM release number is: LLNL-CONF-XXXXXXX